A Class of Feedback-based Coding Algorithms for Broadcast Erasure Channels with Degraded Message Sets

Shirin Bidokhti Saeedi, Marios Gatzianas and Christina Fragouli

Abstract

We consider single-hop broadcast erasure channels (BEC) with degraded message sets and instantaneous feedback regularly available from all receivers, and demonstrate that the main principles of the virtual queue-based algorithms in [1], which were proposed for multiple unicast sessions, can still be applied to this setting and lead to capacity-achieving algorithms. Specifically, we propose a generic class of algorithms and intuitively describe its rationale and properties that result in its efficiency. We then apply this class of algorithms to three examples of BEC channels (with different numbers of users and 2 or 3 degraded message sets) and show that the achievable throughput region matches a known capacity outer bound. These algorithms do not require any prior knowledge of channel statistics for their operation.

I. Introduction

The capacity region of an N-user BEC channel has been determined for the cases of a single multicast session and, recently in [2], [1], for N unicast sessions. Both scenarios can be considered as extremes of a more general N-user setting, in which there exists, for each $\mathcal{S} \subseteq \{1, \ldots, N\}$, a message $W_{\mathcal{S}}$ (equivalently, a set of packets $\mathcal{K}_{\mathcal{S}}$) which is intended for all users in $\mathcal{S}$.

Since the determination of the capacity region under the most general setting is still an open problem, this document studies some special cases, in the hope that the results will provide further insight into the general case, as well as indicate the necessary properties of high-throughput algorithms. Specifically, motivated by the fact that feedback can strictly increase the capacity region of a BEC channel (as has been convincingly demonstrated in [3] for 2 unicast sessions), we consider an N-user BEC with feedback and a two-degraded message set, as well as the special case of a 3-user BEC with 3 degraded messages, and modify the algorithms in [1] to propose capacity-achieving algorithms in this setting.

The algorithms in [1] are recast into a systematic queue-based approach for performing inter-session network coding, which can also be tailored to other problems than the ones studied here (for example, a modification of the algorithm presented for the 3-user BEC with a 3-degraded message set can be used for the 3-user BEC with 2 messages, where message W_1 is intended for users 1,2 while W_2 is intended for user 3). To this end, and in order to illustrate all aspects of this class of algorithms, we present three concrete examples of application.

The problem of degraded multicasting (considered here for the case $N = 3$) has received a lot of attention, since this setting naturally models situations such as multi-layer video transmission, where different users request different quality versions of essentially a single entity (e.g. a video stream). A recent work is [4], where it was shown that an extension of the result by Körner and Marton from two to three users (with a two-message degraded set) is not optimal. As a special case, [5] studied a 3-user system with 2 degraded messages over a combination network whose links are subject to iid erasures and derived its capacity region without feedback. However, in contrast to the current work, these works did not consider the use of feedback.

The document is structured as follows. Section II contains the description of the system model and presents the three examples that will be used to illustrate the proposed class of algorithms, along with a summary of the results. Section III presents the class of algorithms and explains, in intuitive terms, the main ideas behind it, including the important feedback-based actions taken by the transmitter. After a brief summary, in Section IV, of the approach used to derive capacity outer bounds for the BEC channels, Section V presents the exact algorithmic procedure for each of the three selected examples as well as the derivation of the achievable throughput region (inner bound). The latter is seen to match the outer bound in all 3 cases. Section VI concludes this report.

II. Problem formulation

A. System model

We consider a time-slotted single-hop communication system consisting of a single source/trasmitter and N users/receivers with a degraded message set requirement. Denote the set of users with $\mathcal{N} = \{1, \ldots, N\}$. The source

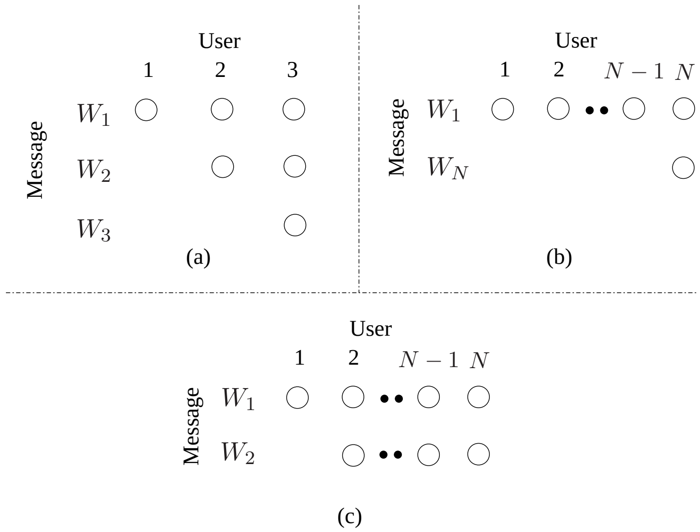

Fig. 1. The message sets of the three channels under investigation.

has N messages $W_1, \ldots, W_N$, where user k wants to receive all messages up to W_k, inclusive. In each slot, the source transmits (i.e. broadcasts) a packet of length L bits (also referred to as "input symbol"; we hereafter use the terms symbol and packet interchangeably) and the channel is modelled as memoryless broadcast erasure so that each broadcast packet/symbol is either received unaltered at a user or is "erased" and is not received by the user. The latter case is equivalent to the user receiving a special symbol (denoted as E), which is distinct from any possible transmitted symbol. At the end of each slot, all users send a simple ACK/NACK reply, through a separate error-free and zero-delay channel, to inform the transmitter whether the packet was received or not.

We denote with $\epsilon_{\{i\}}$ the probability that a packet is erased by user i and with $\epsilon_{\mathcal{S}}$ the probability that a packet is erased by *all* users $k \in \mathcal{S}$.[1] We use the standard information-theoretic definitions [6] of code with feedback, achievable rate and capacity region and, instead of working with messages $W_1, \ldots, W_N$, we assume that there exist sets $\mathcal{K}_1, \ldots, \mathcal{K}_N$ of packets for each corresponding session. We denote with $\mathcal{D}_i = \cup_{j=1}^{i} \mathcal{K}_j$ the set of packets intended for user i and write $K_j = |\mathcal{K}_j|$, for $j \in \mathcal{N}$, where K_j is assumed to be sufficiently large to invoke the law of large numbers. In the sequel, we propose a general class of algorithms in which packets are transmitted until all users have received a sufficient number of linearly independent combinations of their intended packets. Denoting with T^* the average number of transmissions required for this, the rate R_i achieved by the algorithm for message $i = 1, \ldots, N$ can be computed as $R_i = K_i/T^*$. This definition of rate agrees with the commonly used fixed-block length definition of [6], as is shown in [1] through a truncation argument.

B. Summary of main results

We give a full capacity characterization for the degraded message set problem with $N = 3$ and partial capacity characterization when there are N users and only two message sets present. Specifically, we consider the message sets shown in Fig. 1 and prove the following results:

Theorem 1: The capacity region of the 3-user 3-degraded message set (case (a) in Fig. 1) is given by

$$\begin{aligned}\mathcal{C}_{(a)} = \Bigg\{ \boldsymbol{R} \geq \boldsymbol{0} : \frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2+R_3}{1-\epsilon_{\{1,3\}}} \leq 1,\ \frac{R_1+R_2}{1-\epsilon_{\{2\}}} + \frac{R_3}{1-\epsilon_{\{2,3\}}} \leq 1,\ R_1+R_2+R_3 \leq 1-\epsilon_{\{3\}}, \\ \frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2}{1-\epsilon_{\{1,2\}}} + \frac{R_3}{1-\epsilon_{\{1,2,3\}}} \leq 1 \Bigg\},\end{aligned} \tag{1}$$

for arbitrary erasure spatial dependence.

[1]note that the event measured by $\epsilon_{\mathcal{S}}$ does not imply that a packet is erased *only* by the users in $\mathcal{S}$. Other users $j \notin \mathcal{S}$ may also erase the packet.

Algorithm

initialize $Q_{\mathcal{S}}$, $T^i_{\mathcal{S}}$ for all $\mathcal{S} \subseteq \mathcal{N}$ and $i \in \mathcal{S}$;
$t \leftarrow 0$;
for $n \leftarrow 1, \ldots, N$ **do**
  order the queues $Q_{\mathcal{S}}$, with $|\mathcal{S}| = n$, in order of increasing number of non-zero $T^i_{\mathcal{S}}(t)$ (arbitrary tie-breaker);
  for all (ordered $Q_{\mathcal{S}}$ with $|\mathcal{S}| = n$) **do**
    while ($T^i_{\mathcal{S}}(t) > 0$ for at least one $i \in \mathcal{S}$) **do**
      compute *suitable* coefficients $(a_s(p), p \in Q_{\mathcal{T} \supseteq \mathcal{S}})$;
      transmit packet $s = \sum_{p \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_s(p)p$;
      apply `ACTFB` based on received feedback for s;
      $t \leftarrow t + 1$;
    end while
  end for
end for

Fig. 2. Pseudocode for the proposed class of algorithms.

Theorem 2: The capacity regions for the BEC channels with message sets shown in cases (b), (c) of Fig. 1 are given, respectively, by

$$\mathcal{C}_{(b)} = \left\{ \boldsymbol{R} \geq \boldsymbol{0} : \max_{i=1,\ldots,N-1} \left(\frac{R_N}{1 - \epsilon_{\{i,N\}}} + \frac{R_1}{1 - \epsilon_{\{i\}}} \right) \leq 1, \; R_1 + R_N \leq 1 - \epsilon_{\{N\}} \right\}, \tag{2}$$

and

$$\mathcal{C}_{(c)} = \left\{ \boldsymbol{R} \geq \boldsymbol{0} : \max_{i=2,\ldots,N} \left(\frac{R_1}{1 - \epsilon_{\{1\}}} + \frac{R_2}{1 - \epsilon_{\{1,i\}}} \right) \leq 1, \; R_1 + R_2 \leq \min_{i=2,\ldots,N} (1 - \epsilon_{\{i\}}) \right\}, \tag{3}$$

for arbitrary spatial erasure dependence.

III. A CLASS OF ENCODING ALGORITHMS

A. Algorithmic description

We adopt the random linear network coding approach, in which the transmitted packets are viewed as elements of a finite field $\mathbb{F}_q$ (of size q) and the transmitter keeps sending suitable linear combinations of information packets until all users receive a sufficient number of linearly independent combinations with respect to their intended packets. It will be seen that this can be achieved with probability arbitrarily close to 1 for sufficiently large q. We also assume that the issue of conveying the values of the linear combination coefficients to the users is solved through an overhead scheme in the spirit of [1]; the induced overhead is $O(N/L)$. For the reader's convenience, the class of algorithms is next succinctly described in general terms so that it can also be applied to BEC channels with a more general degraded message set request than shown in Fig. 1. We analyze this class of algorithms and characterize the rates it achieves for the BEC channels of Fig. 1 in Section V.

The pseudocode for the proposed class of algorithms is shown in Fig. 2 and explained in more detail below.

1) Virtual queue structure and indices: the source maintains a group of virtual queues $Q_{\mathcal{S}}$, indexed by all non-empty subsets $\mathcal{S} \subseteq \mathcal{N}$, as well as non-negative integers $T^i_{\mathcal{S}}$, for all $\mathcal{S} \subseteq \mathcal{N}$ and $i \in \mathcal{S}$. The above queues and indices are dynamically updated during the algorithm's execution, as will be subsequently described, and should perhaps be denoted as $Q_{\mathcal{S}}(t)$ and $T^i_{\mathcal{S}}(t)$. This explicit time dependence will be omitted in cases it is obvious from context.

2) Initialization: for the case where user $i \in \mathcal{N}$ requests the packets of all sets $\mathcal{K}_1, \ldots, \mathcal{K}_i$, the packets of set $\mathcal{K}_i$ are placed in queue $Q_{\mathcal{N}-\{1,\ldots,i-1\}}$ (equivalently, $Q_{\{i,\ldots,N\}}$).[2] The T indices are initialized as $T^i_{\mathcal{S}}(0) = \|Q_{\mathcal{S}}(0)\|$, where $\|\cdot\|$ denotes the number of packets stored in the queue.

[2]for the most general case where a set of packets $\mathcal{K}_{\mathcal{S}}$ is intended for all users in set $\mathcal{S}$, the packets of $\mathcal{K}_{\mathcal{S}}$ are placed in $Q_{\mathcal{S}}$.

3) Encoding scheme: the source sequentially processes each non-empty queue $Q_{\mathcal{S}}$, for all $\mathcal{S} \subseteq \mathcal{N}$, in order of increasing cardinality $|\mathcal{S}|$; queues with equal $|\mathcal{S}|$ are processed in order of increasing number of non-zero $T_{\mathcal{S}}^{i}$ indices (an abitrary tie-breaker is used for queues with equal set cardinality and equal numbers of non-zero indices). The reason for selecting this order of processing will become apparent soon and further demonstrated in Section V. During the processing of $Q_{\mathcal{S}}$, the source transmits a linear combination of all packets stored in queues $Q_{\mathcal{S}}$ and $Q_{\mathcal{T}}$, for all $\mathcal{T} \supset \mathcal{S}$. We hereafter use the notation $Q_{\mathcal{T} \supseteq \mathcal{S}}$ to refer to this group of queues as a single entity. For example, for $N = 3$, the processing of $Q_{\{1\}}$ consists of transmitting linear combinations of packets in $Q_{\{1\}}$, $Q_{\{1,2\}}$, $Q_{\{1,3\}}$ and $Q_{\{1,2,3\}}$. Note that although multiple queues are involved, we still consider this phase as being "applied" to $Q_{\mathcal{S}}$. The term "suitable" appearing in line 7 of Fig. 2 refers to the fact that the coefficients should be selected in such a way that each user in $\mathcal{S}$ can create a linearly independent equation w.r.t. its packets. More details will be provided in Appendix B (viz. Lemma 5).

4) Feedback-based queue management: while processing queue $Q_{\mathcal{S}}$, the source receives feedback from the users for each transmitted symbol s. Denote with $\mathcal{G}$ the set of users that successfully received s. The source now performs the following actions, collectively referred to as ACTFB:

- if $\mathcal{G} = \varnothing$ or it holds $T_{\mathcal{T}}^{i} = 0$, for all $\mathcal{T} \supseteq \mathcal{S}$ and $i \in \mathcal{S} \cap \mathcal{G}$, then s is retransmitted. Otherwise,
- for each user $i \in \mathcal{S} \cap \mathcal{G}$, find the smallest cardinality set $\hat{\mathcal{S}}_i \supseteq \mathcal{S}$ such that $T_{\hat{\mathcal{S}}_i}^{i} > 0$ (with an arbitrary tie-breaker if more than one such sets exist) and decrease $T_{\hat{\mathcal{S}}_i}^{i}$ by 1.
- if $\mathcal{G} \cap (\mathcal{N} - \mathcal{S}) \neq \varnothing$ and s was erased by at least one user in $\mathcal{S}$, then s is added to queue $Q_{\mathcal{S} \cup \mathcal{G}}$. Additionally, for each $i \in \mathcal{S} - \mathcal{G}$ with $T_{\mathcal{S}}^{i} > 0$, indices $T_{\mathcal{S}}^{i}$, $T_{\mathcal{S} \cup \mathcal{G}}^{i}$ are decreased/increased by 1, respectively.

The second and third items in the above list do not refer to mutually exclusive cases, and both of them may indeed be performed.

5) Condition for stopping the processing of $Q_{\mathcal{S}}$: the source performs steps 3, 4 during the processing of each queue $Q_{\mathcal{S}}$ until it holds $T_{\mathcal{S}}^{i} = 0$ for all $i \in \mathcal{S}$.

B. Algorithmic properties and intuition

The starting place for the analysis of the algorithm's correctness is the following result, which is proved in Appendix A.

Lemma 1: Under the proposed algorithm and for each slot t, any packet p stored in queue $Q_{\mathcal{S}}$ at the beginning of slot t satisfies the following property for all $i \in \mathcal{S}$: p is a linear combination of packets intended for user i (i.e. packets in set $\mathcal{D}_i$) as well as packets received by user i prior to slot t.

It will be useful, for later reference, to rephrase the property in Lemma 1 as follows: under the proposed algorithm, any packet p stored in $Q_{\mathcal{S}}$ at the beginning of slot t can be written, for each $i \in \mathcal{S}$, in the form

$$p = \sum_{u \in \mathcal{D}_i} b_p^{(i)}(u) u + c_p^{(i)}, \tag{4}$$

where $b_p^{(i)}(u)$, $c_p^{(i)}$ are known to user i. Furthermore, the next result follows immediately from (4).

Corollary 1: Consider a set of packets $\mathcal{P}$ such that each $p \in \mathcal{P}$ can be written in the form of (4) for all users in a set $\mathcal{U}$. Then, *any* linear combination $s = \sum_{p \in \mathcal{P}} a_s(p) p$ can also be written in the form of (4) for all users in $\mathcal{U}$.

The reader should notice the conditional nature of the above Corollary. Specifically, the Corollary does *not* guarantee the existence of the set $\mathcal{P}$, having the required property of (4), in the first place; it merely states that, *assuming* the existence of $\mathcal{P}$, each linear combination of packets in $\mathcal{P}$ retains this property. Although (4) was used in [1] to define the notion of "token" for user i, we avoid this terminology here since it may be non-standard. Instead, we define in the next paragraph an "innovative token" for user i as any packet that allows i to create a linearly independent equation for its unknown packets.

Note that when processing $Q_{\mathcal{S}}$ under the proposed algorithm, the transmitted packet $s = \sum_{p \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_s(p) p$ is a linear combination of all packets stored in queues $Q_{\mathcal{T}}$, with $\mathcal{T} \supseteq \mathcal{S}$. Combining Lemma 1 (which states that each packet stored in $Q_{\mathcal{T}}$ can be written in the form of (4) for each $i \in \mathcal{T}$) with Corollary 1, we conclude that s can be written according to (4) for all $i \in \mathcal{S}$. Hence, with a suitable selection of coefficients $a_s(p)$, a successfull reception of s by user i allows i to create, through (4), a linearly independent equation for its packets. This motivates the following definition.

Definition 1: A packet s is an innovative token for user i if the successfull reception of s by i allows i to create a linear equation for its unknown packets in $\mathcal{D}_i$ that is linearly independent w.r.t. previously created equations by i.

Hence, each user $i \in \mathcal{N}$ can decode its packets in a one-shot manner after receiving $|\mathcal{D}_i|$ innovative tokens. The following result, proved in Appendix B, shows that this can indeed be achieved for a sufficiently large field size.

Lemma 2: Under the proposed algorithm and for $q > N$, the coefficients of the linear combinations transmitted in each slot can be selected in such a way that each user $i \in \mathcal{N}$ receives $|\mathcal{D}_i|$ innovative tokens by the end of the algorithm.

Lemma 2 guarantees the correctness of the proposed algorithm. Although an inspection of its proof will reveal some of the intuition behind the algorithm, we also provide the following intuitive discussion. A packet s that is transmitted from the source at slot t can be "useful" for user i, if received by i, in two different ways:

(1) At the time of transmission (i.e. before s is actually transmitted), packet s is an innovative token for user i, so that user i creates an equation, which is linearly independent of previously created equations at user i. In this sense, s offers an immediate "benefit" to user i.

(2) At the time of transmission, packet s is not an innovative token for user i, but since it is received at user i, it can offer side information to user i. In this sense, packet s can be combined in the future with a side information packet of another user and create a packet that is useful to both of them, thus offering a "delayed benefit".

We give an example for these two types of useful information. A packet $p_3 \in \mathcal{K}_3$ offers immediate benefit only to user 3. Nonetheless, p_3 can also offer delayed benefit as follows: if p_3 is only received at user 2 (which we denote with p_3^2, so that the upper index indicates the user who received the packet), it can potentially be combined with a packet of the form $p_2 \in \mathcal{K}_2$ later on, so that the resulting linear combination $\langle p_2, s_3^2 \rangle$ allows both users 2 and 3 create an equation upon successful reception. Clearly, this efficient mixing of side information requires the transmitter to know exactly which user received which packets; this knowledge is acquired through feedback.

Based on the above, the most "useful" type of transmitted packet is a linear combination of packets, each of which can be written in the form of (4) for *all* users $i \in \mathcal{N}$, since such a packet can, by properly selecting the coefficients, allow all N users to create linear equations upon its reception. Clearly, before any transmissions occur and when no side information has been received, each packet can be written in the form of (4) only w.r.t. the users for which it is destined. Hence, the most efficient packets, at the beginning of the algorithm, belong to $\mathcal{K}_1$. However, as transmissions are scheduled and side information is created, additional packets may become "useful" for all N users. For example, for the case $N = 3$, a packet $p_2 \in \mathcal{K}_2$ that is received by 1 can be written in the form of (4) for all 3 users.

The indices $T^i_{\mathcal{S}}$ are interpreted as the number of linearly independent combinations that user i still needs to receive during the processing of $Q_{\mathcal{S}}$ in step 3 of Section III-A. When it holds $T^i_{\mathcal{S}} = 0$, then user i has gathered all available information from $Q_{\mathcal{S}}$ and the queue is no longer useful for i, though it is still useful for other users $j \in \mathcal{S}$ with $T^j_{\mathcal{S}} > 0$. Processing of $Q_{\mathcal{S}}$ therefore stops, as explained, only when *all* $T^i_{\mathcal{S}}$, $i \in \mathcal{S}$, become 0.

The efficiency of the proposed algorithm lies in the fact that it tries to "exploit" each slot as much it can, in the sense that the transmitted packet potentially allows each user, upon successful reception of the packet, to either create an equation or gain side information. While processing $Q_{\mathcal{S}}$, packets s are being transmitted that satisfy the property in Lemma 1 for all users in $\mathcal{S}$. When s is received by a user $i \in \mathcal{S}$, this user forms an equation for its unknown packets. When s is, however, received by a user $i \notin \mathcal{S}$ (which cannot currently create an equation from s), the packet can still be useful, if properly handled. ACTFB tries to optimally handle this based on the feedback received and according to the following maximality principle:

Maximal tokeness (MT): a transmitted packet s should be placed in $Q_{\mathcal{S}'}$ iff $\mathcal{S}'$ is the maximal set such that s can be written in the form of (4) for *all* users in $\mathcal{S}'$.

Notice that the queue initialization also conforms to the MT rule. Since we have qualitatively defined the "efficiency" of a packet as the number of users for which it can be decomposed as in (4), we conclude that enforcing the MT principle results in a packet becoming more efficient as it is stored in queues $Q_{\mathcal{S}}$ of increasing $|\mathcal{S}|$.

Interpretation of ACTFB: The three steps of ACTFB follow naturally from the MT principle and the interpretation of indices $T^i_{\mathcal{S}}$. Specifically, step 1 captures the fact that it is possible for a packet s to be received by no user, in which case s is retransmitted, since it still contains useful information. The second condition in step 1 of ACTFB1 is more intriguing but can be simply restated as follows: if a packet s, which is a linear combination of all packets in $Q_{\mathcal{T}\supseteq S}$ is received *only* by users which have already recovered all available equations in *all* queues $Q_{\mathcal{T}\supseteq S}$ (i.e. $T^i_{\mathcal{T}} = 0$ for all $\mathcal{T} \supseteq \mathcal{S}$), then the packet is retransmitted. Although this may lead to inefficiency, the analysis performed in Section V indicates that this does happen for the message sets in Fig. 1 (i.e. the algorithm achieves a capacity outer bound). The question of whether this property holds for general N and arbitrary degraded messages is an open problem.

Step 2 merely says that a user $i \in \mathcal{S}$ that receives s can construct an equation for its unknown packets (due to Lemma 1 and Corollary 1). Through a proper selection of coefficients (see the proof of Lemma 2), this equation is linearly independent w.r.t. all equations previously created by i. Hence, the corresponding counter $T^i_{\mathcal{S}}$ should be decreased by 1 to capture this fact.

Finally, step 3 corresponds to the case where packet s is not received by all receivers which can create equations from it (i.e., all users in $\mathcal{S}$) but is received by some users that do not belong to $\mathcal{S}$ (i.e., packet s cannot be written in the form of (4) for these users at the time of transmission). After reception of this packet, since the packet is now known to all users in $\mathcal{G} - \mathcal{S}$, the packet can be decomposed according to (4) for all users in $\mathcal{S} \cup \mathcal{G}$, the latter set being maximal. Hence, by the MT rule, s should be placed in $Q_{\mathcal{S}\cup\mathcal{G}}$ for the next time slot.

Furthermore, although there might be some user $j \in \mathcal{S}$ that did not receive packet s (i.e. $j \in \mathcal{S} - \mathcal{G}$), this information can be sent in the future through linear combinations transmitted from queue $Q_{\mathcal{S}\cup\mathcal{G}}$ (where packet s was moved). In fact, it is more efficient for user $j \in \mathcal{S} - \mathcal{G}$ to receive a linear combination (containing s) from $Q_{\mathcal{S}\cup\mathcal{G}}$ rather than s itself from $Q_{\mathcal{S}}$, since the former combination can provide information to more users. This is modeled by decreasing/increasing $T^i_{\mathcal{S}}$, $T^i_{\mathcal{S}\cup\mathcal{G}}$ by one.

Remark 1: Since the processing of $Q_{\mathcal{S}}$ may place some packets in $Q_{\mathcal{S}'}$, with $\mathcal{S}' \supset \mathcal{S}$ (as well as increase some $T^i_{\mathcal{S}'}$) and since all queues with non-zero T indices have to be processed eventually, selecting the order of processing according to increasing $|\mathcal{S}|$ means that each queue $Q_{\mathcal{S}}$ will be processed, according to step 3 of Section III-A, in only one stage. Selecting a different order such that, for two sets $\mathcal{S}_1, \mathcal{S}_2$ with $\mathcal{S}_1 \supset \mathcal{S}_2$, queue $Q_{\mathcal{S}_1}$ is processed before $Q_{\mathcal{S}_2}$ allows for the possibility that $Q_{\mathcal{S}_1}$ may have to be processed twice, the second time being due to packets newly moved from $Q_{\mathcal{S}_2}$. To avoid this issue, which would make the analysis of the algorithm more difficult, we stick to processing the queues in order of increasing $|\mathcal{S}|$.

IV. Capacity outer bounds

In this Section, we provide capacity outer bounds, for each of the 3 cases shown in Fig. 1, that are tighter than the well-known corresponding cut-set bounds $\mathcal{C}_{cs}$. The derivation of these bounds follows the often-used procedure [7], [3], [2] of introducing additional auxiliary links of infinite capacity among the users to create physically degraded channels, whose capacity (with or without feedback) is achieved through simple timesharing among the users. We illustrate this in detail for the 3 user 3-message degraded set (case (a) in Fig. 1) and only provide the main points for the other cases in Fig. 1. The next Section contains the analysis of the algorithm, as applied to each case in Fig. 1, and presents corresponding matching inner bounds.

A. 3-message degraded set: case (a) in Fig. 1

We consider the auxiliary channels $\hat{C}, \tilde{C}, \mathring{C}$ shown in Fig. 3, where dashed lines indicate infinite capacity links. The cut-set bound for channel C is written as $\mathcal{C}_{cs} = \left\{ \boldsymbol{R} \geq \boldsymbol{0} : \sum_{k=1}^{i} R_k \leq 1 - \epsilon_{\{i\}},\ i = 1, \ldots, 3 \right\}$.

It is not difficult to see that any encoding scheme at the source that achieves rates R_1, R_2 and rate R_3 over the original erasure channel C, could be used over channel $\hat{C}$ to reliably communicate messages W_2 and W_3 to user 3 and message W_1 to user 1. Similarly, any such code could be used over channel $\tilde{C}$ to reliably communicate messages W_1 and W_2 to user 2 and message W_3 to user 3. Finally, the same code could be used over channel $\mathring{C}$ to communicate message W_i to user i for $i = 1, 2, 3$. Therefore, these 3 unicast problems give outer bounds on the capacity region of our interest; i.e., denoting with $\hat{\mathcal{C}}$, $\tilde{\mathcal{C}}$, $\mathring{\mathcal{C}}$ the capacity regions of the aforementioned unicast problems over $\hat{C}$, $\tilde{C}$, $\mathring{C}$ we get

$$\mathcal{C}_{(a)} \subseteq \mathcal{C}_{cs} \cap \hat{\mathcal{C}} \cap \tilde{\mathcal{C}} \cap \mathring{\mathcal{C}}. \tag{5}$$

Over channel $\hat{C}$, a symbol is actually erased at user 3 in channel $\hat{C}$ iff both users 1, 3 erase the symbol in channel C, while user 2 erases the symbol in channel $\hat{C}$ iff all three users erase it in channel C. This implies that

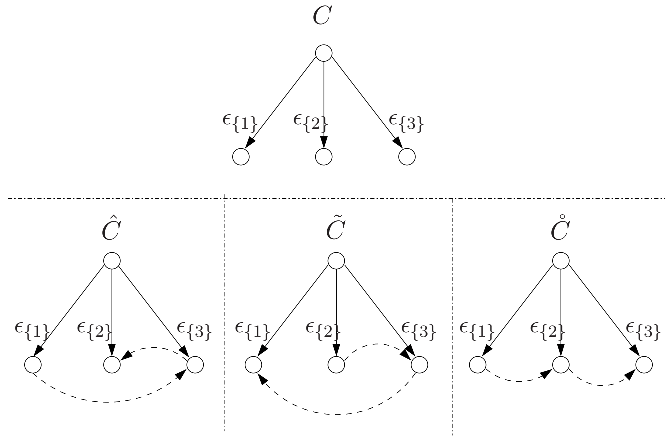

Fig. 3. Channel C shows a BEC with $N = 3$ users. Channels $\hat{C}$, $\tilde{C}$, $\mathring{C}$ are the auxiliary channels used to derive capacity outer bounds for the three-message set problem.

$\hat{C}$ is also a BEC with parameters $\hat{\epsilon}_{\{1\}} = \epsilon_{\{1\}}$, $\hat{\epsilon}_{\{2\}} = \epsilon_{\{1,2,3\}}$ and $\hat{\epsilon}_{\{3\}} = \epsilon_{\{1,3\}}$. Analogous conclusions can be drawn for channel $\tilde{C}$, $\mathring{C}$.

Furthermore, since channels $\hat{C}$, $\tilde{C}$, $\mathring{C}$ are physically degraded [8], we have the following well-known results:

- feedback does not increase the capacity of a physically degraded channel [9].
- the capacity region (without feedback) of the N-user physically degraded BEC is achieved through timesharing among the users [10]. For example, the capacity region for channel $\hat{C}$ is given by

$$\hat{\mathcal{C}} = \left\{ \boldsymbol{R} \geq \boldsymbol{0} : \frac{R_1}{1-\hat{\epsilon}_1} + \frac{R_2 + R_3}{1-\hat{\epsilon}_3} \leq 1 \right\} = \left\{ \boldsymbol{R} \geq \boldsymbol{0} : \frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2 + R_3}{1-\epsilon_{\{1,3\}}} \leq 1 \right\},$$

  with analogous expressions for $\tilde{\mathcal{C}}$, $\mathring{\mathcal{C}}$.

Combining the above to evaluate $\hat{\mathcal{C}}$, $\tilde{\mathcal{C}}$ and $\mathring{C}$ in (5), the RHS of (5) matches (1) of Theorem 1 and yields a tighter capacity outer bound than $\mathcal{C}_{cs}$.

B. 2-message degraded set: case (b) in Fig. 1

The cut-set bound for this system is immediately written as $\mathcal{C}_{cs} = \{(R_1, R_N) \geq \boldsymbol{0} : R_1 \leq \min_{i=1,\ldots,N-1}(1 - \epsilon_{\{i\}}),\ R_1 + R_N \leq 1 - \epsilon_{\{N\}}\}$. A tighter bound can be derived using the auxiliary channel $\hat{C}_i$ shown in the left side of Fig. 4 (again, dashed lines indicate infinite capacity links). Specifically, a coding scheme that achieves rates R_1, R_N in C also achieves unicast rates R_1 and R_N for users i and N, respectively, in channel $\hat{C}_i$. Denoting with $\hat{\mathcal{C}}_i$ the capacity region of this physically degraded channel, it holds

$$\hat{\mathcal{C}}_i = \left\{ (R_1, R_N) \geq \boldsymbol{0} : \frac{R_1}{1-\epsilon_{\{i\}}} + \frac{R_N}{1-\epsilon_{\{i,N\}}} \leq 1 \right\}. \tag{6}$$

Notice that there exist $N-1$ such channels $\hat{C}_i$, one for each $i \neq N$. Hence, a tighter outer capacity bound for C has the form

$$\mathcal{C}_{cs} \cap \bigcap_{i=1}^{N-1} \hat{\mathcal{C}}_i = \left\{ (R_1, R_N) \geq \boldsymbol{0} : R_1 + R_N \leq 1 - \epsilon_{\{N\}},\ \max_{i=1,\ldots,N-1} \left(\frac{R_1}{1-\epsilon_{\{i\}}} + \frac{R_N}{1-\epsilon_{\{i,N\}}} \right) \leq 1 \right\}, \tag{7}$$

which matches (2).

C. 2-message degraded set: case (c) in Fig. 1

The cut-set bound in this case is given by $\mathcal{C}_{cs} = \{(R_1, R_2) \geq \boldsymbol{0} : R_1 \leq 1-\epsilon_{\{1\}},\ R_1+R_2 \leq \min_{i=2,\ldots,N} \left(1-\epsilon_{\{i\}}\right)\}$. We next consider the auxiliary channel $\hat{C}_i$ shown in the right side of Fig. 4 and note that any coding scheme that

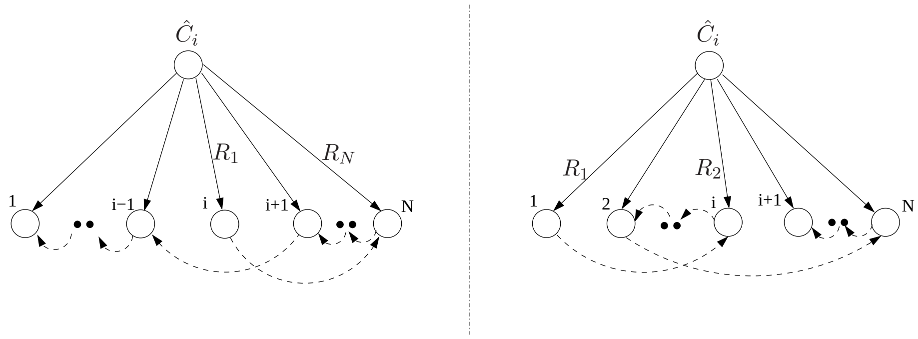

Fig. 4. Auxiliary channels used for deriving capacity outer bounds of case (b) (left) and case (c) (right) in Fig. 1.

achieves a rate (R_1, R_2) in C also achieves unicast rates R_1 and R_2 for users 1 and i, respectively, in $\hat{C}_i$. Since this channel is physically degraded as well, the capacity region $\hat{\mathcal{C}}_i$ for $\mathcal{C}_i$ is given by

$$\hat{\mathcal{C}}_i = \left\{ (R_1, R_2) \geq \mathbf{0} : \frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2}{1-\epsilon_{\{1,i\}}} \leq 1 \right\}. \tag{8}$$

There exist $N-1$ channels $\hat{C}_i$, one for each $i \neq 1$. Using a similar argument as in the previous section, a tighter capacity outer bound for C is

$$\mathcal{C}_{cs} \cap \bigcap_{i=2}^{N} \hat{\mathcal{C}}_i = \left\{ (R_1, R_2) \geq \mathbf{0} : R_1 + R_2 \leq \min_{i=2,\dots,N} (1-\epsilon_{\{i\}}),\ \max_{i=2,\dots,N} \left(\frac{R_1}{1-\epsilon_{\{1\}}} + \frac{R_2}{1-\epsilon_{\{1,i\}}} \right) \leq 1 \right\}, \tag{9}$$

which matches (3).

V. Achievability results

In this Section, we present the derivation of capacity inner bounds, as achieved under the application of the proposed algorithm to each of the cases in Fig. 1. These inner bounds will be seen to match the corresponding outer bounds of the previous Section.

A. 3-message degraded set: case (a) in Fig. 1

The algorithm uses all queues $Q_{\mathcal{S}}$, with $\mathcal{S} \subseteq \{1,2,3\}$, except for $Q_{\{1,2\}}$, which remains empty for the entire duration of the algorithm. The packets of sets $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ are placed into queues $Q_{\{1,2,3\}}$, $Q_{\{2,3\}}$, $Q_{\{3\}}$, respectively, and the T indices are initialized according to step 2 of Section III-A as follows: $T^3_{\{3\}}(0) = K_3$, $T^2_{\{2,3\}}(0) = T^3_{\{2,3\}}(0) = K_2$, and $T^1_{\{1,2,3\}}(0) = T^2_{\{1,2,3\}}(0) = T^3_{\{1,2,3\}}(0) = K_1$. All other T indices are set to 0. The source then processes the queues in the following order: $Q_{\{3\}}$, $Q_{\{1,3\}}$, $Q_{\{2,3\}}$, $Q_{\{1,2,3\}}$. This order is dictated by the rule (see step 3 of Section III-A) of processing queues $Q_{\mathcal{S}}$ in order of increasing $|\mathcal{S}|$ and the fact that it holds $T^1_{\{1,3\}} = 0$ for the entire algorithm's execution. The latter fact follows from the construction of the algorithm and `ACTFB`, which implies that $T^1_{\{1,3\}}$ can only be increased during the processing of queue $Q_{\{1\}}$, which is empty by construction and is, therefore, never processed. Hence, $Q_{\{1,3\}}$ has fewer non-zero T indices than $Q_{\{2,3\}}$ and is processed first.

Recall that the processing of $Q_{\mathcal{S}}$ entails transmitting linear combinations from all queues $Q_{\mathcal{T} \supseteq \mathcal{S}}$ until all $T^i_{\mathcal{S}}$ become 0. We denote with $T^*_{\mathcal{S}}$ the number of slots required for processing queue $Q_{\mathcal{S}}$, and with $T^i_{\mathcal{S}}(t)$ the value of $T^i_{\mathcal{S}}$ at slot t. Hence, it follows that

$$T^*_3 = \frac{T^3_{\{3\}}(0)}{1-\epsilon_{\{1,2,3\}}} = \frac{K_3}{1-\epsilon_{\{1,2,3\}}}, \tag{10}$$

since any packet that is received by at least one user leads, due to step 3 of `ACTFB1`, to a reduction of $T^3_{\{3\}}$ by 1.

The values of the indices at the end of processing $Q_{\{3\}}$ (denote this epoch as t_3) are

$$\begin{aligned} T^3_{\{2,3\}}(t_3) &= K_2 + T^*_{\{3\}}(\epsilon_{\{1,3\}} - \epsilon_{\{1,2,3\}}), \\ T^3_{\{1,3\}}(t_3) &= T^*_{\{3\}}(\epsilon_{\{2,3\}} - \epsilon_{\{1,2,3\}}), \quad T^1_{\{1,3\}}(t_3) = 0, \\ T^3_{\{1,2,3\}}(t_3) &= K_1 + T^*_{\{3\}}(\epsilon_{\{3\}} - \epsilon_{\{1,3\}} - \epsilon_{\{2,3\}} + \epsilon_{\{1,2,3\}}), \\ T^2_{\{2,3\}}(t_3) &= K_2, \quad T^1_{\{1,2,3\}}(t_3) = T^2_{\{1,2,3\}}(t_3) = K_1, \end{aligned} \tag{11}$$

and follow again from the logic of step 3 of ACTFB (the terms inside parentheses express the probability that a packet is only seen by a specific subset of the 3 users). For example, the second term in the RHS of the expression for $T^3_{\{2,3\}}(t_3)$ captures the fact that, due to step 3 of ACTFB, index $T^3_{\{2,3\}}$ can be increased if, during the processing of $Q_{\{3\}}$, a packet is received only by 2 (and not 1,3). The probability of this event appears inside the parentheses and, since $Q_{\{3\}}$ is processed for $T^*_{\{3\}}$ slots, the interpretation is obvious. A similar explanation can be given for the other relations in (11).

To make the equations more compact, we denote with $\Delta^+_{\mathcal{S}} T^i_{\mathcal{T}}$, $\Delta^-_{\mathcal{S}} T^i_{\mathcal{T}}$ the total increase/decrease, respectively, of index $T^i_{\mathcal{T}}$ when the algorithm processes $Q_{\mathcal{S}}$, with $\mathcal{S} \subset \mathcal{T}$. Using this notation, we can write, for example, $\Delta^+_{\{3\}} T^3_{\{2,3\}} = T^*_{\{3\}}(\epsilon_{\{1,3\}} - \epsilon_{\{1,2,3\}})$. The source next processes $Q_{\{1,3\}}$ for a total of

$$T^*_{\{1,3\}} = \frac{T^3_{\{1,3\}}(t_3)}{1 - \epsilon_{\{2,3\}}}, \tag{12}$$

time slots (since $T^3_{\{1,3\}}$ is reduced if the transmitted packet is received by either 2 or 3), while the indices $T^3_{\{1,2,3\}}$, $T^1_{\{1,2,3\}}$ are modified, due to step 3 of ACTFB as follows

$$\begin{aligned} \Delta^+_{\{1,3\}} T^3_{\{1,2,3\}} &= T^*_{\{1,3\}}(\epsilon_{\{3\}} - \epsilon_{\{2,3\}}), \\ \Delta^-_{\{1,3\}} T^1_{\{1,2,3\}} &= T^*_{\{1,3\}}(1 - \epsilon_{\{1\}}). \end{aligned} \tag{13}$$

Notice that the second equation in (13) is due to the fact that $T^1_{\{1,3\}}$ is already 0 at this stage so that, according to step 3 of ACTFB, the source should find the smallest cardinality $\hat{\mathcal{S}} \supset \{1,3\}$ such that $T^1_{\hat{\mathcal{S}}} > 0$. In this case, there exists only one such index, namely $T^1_{\{1,2,3\}}$. Furthermore, if $T^1_{\{1,2,3\}}$ becomes zero before the processing of $Q_{\{1,3\}}$ is complete, then, according to Lemma 3 in Appendix B, user 1 has received enough linear combinations to decode its packets.

The source next processes $Q_{\{2,3\}}$ for a total of

$$T^*_{\{2,3\}} = \max\left(\frac{T^2_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,2\}}}, \frac{T^3_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,3\}}}\right), \tag{14}$$

time slots. The modification of the indices in $Q_{\{1,2,3\}}$ during this stage is

$$\begin{aligned} \Delta^+_{\{2,3\}} T^3_{\{1,2,3\}} &= \frac{T^3_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,3\}}}(\epsilon_{\{3\}} - \epsilon_{\{1,3\}}), \\ \Delta^+_{\{2,3\}} T^2_{\{1,2,3\}} &= \frac{T^2_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,2\}}}(\epsilon_{\{2\}} - \epsilon_{\{1,2\}}), \end{aligned} \tag{15}$$

and

$$\begin{aligned} \Delta^-_{\{2,3\}} T^3_{\{1,2,3\}} &= \left(T^*_{\{2,3\}} - \frac{T^3_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,3\}}}\right)(1 - \epsilon_{\{3\}}), \\ \Delta^-_{\{2,3\}} T^2_{\{1,2,3\}} &= \left(T^*_{\{2,3\}} - \frac{T^2_{\{2,3\}}(t_3)}{1 - \epsilon_{\{1,2\}}}\right)(1 - \epsilon_{\{2\}}). \end{aligned} \tag{16}$$

Although (15) can be interpreted similarly to (11), the explanation for (16) is slightly more involved. Specifically, consider the ACTFB actions taken w.r.t. user 3 when processing $Q_{\{2,3\}}$. As long as it holds $T^3_{\{2,3\}} > 0$, any transmitted packet that is received by user 1 and erased by user 3 leads, through step 3 of ACTFB, to a decrease of

$T^3_{\{2,3\}}$ and a corresponding increase of $T^3_{\{1,2,3\}}$, which is captured in (15). Additionally, once it holds $T^3_{\{2,3\}} = 0$, any transmitted packet received by user 3 leads to a decrease of $T^3_{\{1,2,3\}}$. Since it takes $T^3_{\{2,3\}}(t_3)/(1-\epsilon_{\{1,3\}})$ slots for $T^3_{\{2,3\}}$ to become 0, it follows that $T^3_{\{1,2,3\}}$ can be decreased only in the remaining $T^*_{\{2,3\}} - T^3_{\{2,3\}}(t_3)/(1-\epsilon_{\{1,3\}})$, slots whence the first relation in (16) follows.

Hence, at the end of processing $Q_{\{2,3\}}$ (denote this time with t_{23}), the indices in $Q_{\{1,2,3\}}$ have the values

$$T^i_{\{1,2,3\}}(t_{23}) = \left[T^i_{\{1,2,3\}}(t_3) + \sum_{\mathcal{S}} \Delta^+_{\mathcal{S}} T^i_{\{1,2,3\}} - \sum_{\mathcal{S}} \Delta^-_{\mathcal{S}} T^i_{\{1,2,3\}}\right]^+, \tag{17}$$

where the summation is performed over $\mathcal{S} \in \{\{1,3\},\{2,3\}\}$ and $[x]^+ = \max(x,0)$. Therefore, the processing of $Q_{\{1,2,3\}}$ by itself requires

$$T^*_{\{1,2,3\}} = \max_{i=1,\dots,3} \frac{T^i_{\{1,2,3\}}(t_{23})}{1-\epsilon_{\{i\}}}, \tag{18}$$

slots. Denoting the sum of the slots for all phases as T^*, i.e. $T^* = T^*_{\{3\}} + T^*_{\{1,3\}} + T^*_{\{2,3\}} + T^*_{\{1,2,3\}}$ the algorithm achieves a rate of $R_j = K_j/T^*$ for $j = 1,2,3$. Simple algebra (which is conveniently performed by symbolic manipulation packages) now reveals that the achievable throughput region exactly matches (1), which, combined with the discussion in Section IV-A, yields the full characterization of the capacity region. Notice that no assumption on spatially independent erasures was made, so that the result holds for arbitrary erasures.

B. 2-message degraded set: case (b) in Fig. 1

The algorithm only operates on queues $Q_{\mathcal{S}}$ such that $N \in \mathcal{S}$ and initially places the packets of sets $\mathcal{K}_1$, $\mathcal{K}_N$ into $Q_{\mathcal{N}}$, $Q_{\{N\}}$, respectively. It also initializes the T indices as $T^N_{\{N\}} = K_N$ and $T^i_{\mathcal{N}} = K_1$ for all $i \in \mathcal{N}$, while all other indices are set to 0. Additionally, since, for all $\mathcal{S} \subset \mathcal{N}$ and all $i \neq N$, the indices $T^i_{\mathcal{S}}$ are 0 for the entire duration of the algorithm (users $1,\dots,N-1$ only require packets from $Q_{\mathcal{N}}$), the encoding scheme in step 3 of Section III-A can be simplified by combining $Q_{\mathcal{S}}$ directly with $Q_{\mathcal{N}}$.

For each $\mathcal{S} \subset \mathcal{N}$, we denote with $t_{\mathcal{S}}$, $T^*_{\mathcal{S}}$, respectively, the slot when the processing of $Q_{\mathcal{S}}$ *begins* and the number of slots required for the entire processing of $Q_{\mathcal{S}}$. By construction of the algorithm it holds

$$T^*_{\mathcal{S}} = \frac{T^N_{\mathcal{S}}(t_{\mathcal{S}})}{1-\epsilon_{\mathcal{N}-(\mathcal{S}-\{N\})}} \quad \mathcal{S} \subseteq \mathcal{N}, \tag{19}$$

since there is only one non-zero T index (namely, $T^N_{\mathcal{S}}$) in each $Q_{\mathcal{S}}$, and this index is decreased by 1, due to step 3 of ACTFB, whenever the transmitted packet is received by either user N or by any user outside set $\mathcal{S} - \{N\}$.

We introduce the notation $p_{\mathcal{A},\mathcal{B}}$, for disjoint sets $\mathcal{A}$, $\mathcal{B}$, to denote the probability that a transmitted packet is erased by *all* users in $\mathcal{A}$ and received by *all* users in $\mathcal{B}$. Defining

$$k^N_{\mathcal{S}} = \begin{cases} T^N_{\mathcal{S}}(t_{\mathcal{S}}) & \text{for} \quad \mathcal{S} \subset \mathcal{N}, \\ T^N_{\mathcal{N}}(t_{\mathcal{N}}) - K_1 & \text{for} \quad \mathcal{S} = \mathcal{N}, \end{cases} \tag{20}$$

where $t_{\mathcal{N}}$ is the slot that corresponds to the beginning of processing $Q_{\mathcal{N}}$, the following recursive relation can be derived, based on the logic of step 3 of ACTFB.

$$k^N_{\mathcal{S}} = \sum_{\substack{\varnothing \neq \mathcal{I} \subset \mathcal{S} \\ N \in \mathcal{I}}} \frac{k^N_{\mathcal{I}}}{1-\epsilon_{\mathcal{N}-(\mathcal{I}-\{N\})}} p_{\mathcal{N}-(\mathcal{S}-\{N\}),\mathcal{S}-\mathcal{I}} \quad \forall \mathcal{S} \subseteq \mathcal{N}. \tag{21}$$

The above recursion may seem cryptic at first but it merely states the fact that $k^N_{\mathcal{S}}$ (i.e. the value of $T^N_{\mathcal{S}}$ at the beginning of processing $Q_{\mathcal{S}}$, for $\mathcal{S} \subset \mathcal{N}$) is equal to the cumulative increase of $T^N_{\mathcal{S}}$, due to step 3 of ACTFB, when processing all queues $Q_{\mathcal{I}}$ from which a packet may potentially move to $Q_{\mathcal{S}}$. Especially for $k^N_{\mathcal{N}}$, we need to take into account the fact that $T^N_{\mathcal{N}}$ was initialized to K_1, so that $T^N_{\mathcal{N}}(t_{\mathcal{N}})$ is the sum of the cumulative increase $k^N_{\mathcal{N}}$ and K_1. Thankfully, the solution of the recursion in (21) has been provided in [1], whence we take the following result:

$$k^N_{\mathcal{S}} = K_N \left(1-\epsilon_{\mathcal{N}-(\mathcal{S}-\{N\})}\right) \sum_{\mathcal{H} \subseteq \mathcal{S}-\{N\}} \frac{(-1)^{|\mathcal{S}|-|\mathcal{H}|-1}}{1-\epsilon_{\mathcal{N}-\mathcal{H}}} \quad \forall \mathcal{S} \supseteq \{N\}. \tag{22}$$

Hence, the state of the $T_{\mathcal{N}}^i$ indices at $t_{\mathcal{N}}$ is as follows

$$\begin{aligned} T_{\mathcal{N}}^N(t_{\mathcal{N}}) &= K_1 + k_{\mathcal{N}}^N, \\ T_{\mathcal{N}}^i(t_{\mathcal{N}}) &= K_1 - \sum_{\{i,N\}\subseteq\mathcal{S}\subset\mathcal{N}} T_{\mathcal{S}}^*(1-\epsilon_{\{i\}}), \quad i = 1,\dots,N-1, \end{aligned} \tag{23}$$

where the second relation is due to the fact that, when combining $Q_{\mathcal{S}}$ with $Q_{\mathcal{N}}$ for $i \in \mathcal{S}$, any packet received by $i \neq N$ leads to a decrease of $T_{\mathcal{N}}^i$ by 1, due to step 3 of ACTFB. The number of slots required for processing $Q_{\mathcal{N}}$ is

$$T_{\mathcal{N}}^* = \max_{i\in\mathcal{N}} \frac{T_{\mathcal{N}}^i}{1-\epsilon_{\{i\}}}. \tag{24}$$

Hence, the total number of slots required by the algorithm is $T^* = \sum_{\{N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} T_{\mathcal{S}}^*$, whence the rate $R_j = K_j/T^*$ for $j = 1, N$ can be computed. The detailed computations are provided in Appendix C and show that the achievable region is given by (3). This result holds for arbitrary erasure spatial dependence.

C. 2-message degraded set: case (c) in Fig. 1

This is the simplest case of the 3 examined, in the sense that only two queues are required, namely $Q_{\mathcal{N}}$ and $Q_{\mathcal{N}-\{1\}}$. The packets of sets $\mathcal{K}_1, \mathcal{K}_2$ are placed into $Q_{\mathcal{N}}$, $Q_{\mathcal{N}-\{1\}}$, respectively, along with the suitable initialization: $T_{\mathcal{N}-\{1\}}^i(0) = K_2$ for $i \neq 1$ and $T_{\mathcal{N}}^i(0) = K_1$ for $i \in \mathcal{N}$. The source first combines queues $Q_{\mathcal{N}-\{1\}}$ and $Q_{\mathcal{N}}$ until all $T_{\mathcal{N}-\{1\}}^i$ become 0 and then processes $Q_{\mathcal{N}}$ by itself.

Thinking in similar lines as for the analysis of the 3-message degraded set, the number of slots $T_{\mathcal{N}-\{1\}}^*$ required to process $Q_{\mathcal{N}-\{1\}}$ is

$$T_{\mathcal{N}-\{1\}}^* = \max_{i\in\mathcal{N}-\{1\}} \frac{T_{\mathcal{N}-\{1\}}^i(0)}{1-\epsilon_{\{1,i\}}} = \max_{i\in\mathcal{N}-\{1\}} \frac{K_2}{1-\epsilon_{\{1,i\}}}, \tag{25}$$

since $T_{\mathcal{N}-\{1\}}^i$ is decreased if the transmitted packet is received by either i or any user outside $\mathcal{N}-\{1\}$ (i.e. user 1). The cumulative increase and decrease of $T_{\mathcal{N}}^i$ can be computed similarly to (15)–(16) as

$$\begin{aligned} \Delta_{\mathcal{N}-\{1\}}^+ T_{\mathcal{N}}^i &= \frac{K_2}{1-\epsilon_{\{1,i\}}}(\epsilon_{\{i\}} - \epsilon_{\{1,i\}}) \quad i-2,\dots N, \\ \Delta_{\mathcal{N}-\{1\}}^- T_{\mathcal{N}}^i &= \left(T_{\mathcal{N}-\{1\}}^* - \frac{K_2}{1-\epsilon_{\{1,i\}}}\right)(1-\epsilon_{\{i\}}) \quad i = 2,\dots,N, \end{aligned} \tag{26}$$

so that at the end of processing $Q_{\mathcal{N}-\{1\}}$ (denote this time with t_1) it holds $T_{\mathcal{N}}^1(t_1) = K_1$ and

$$T_{\mathcal{N}}^i(t_1) = \left[T_{\mathcal{N}}^i(0) + \Delta_{\mathcal{N}-\{1\}}^+ T_{\mathcal{N}}^i - \Delta_{\mathcal{N}-\{1\}}^- T_{\mathcal{N}}^i\right]^+, \quad i = 2,\dots,N. \tag{27}$$

The rest of the analysis follows the lines of Section V-A and reveals, after some simple algebra, that the proposed algorithm achieves the region of (3), which matches the outer bound of Section IV-C. Notice that, again, this result holds for arbitrary erasure spatial dependence.

VI. Conclusions

This report revisited the virtual-queue based coding algorithm proposed in [1] for BEC channels with multiple unicast sessions and demonstrated that its main concepts of token handling and keeping track of the number of linearly independent equations required by each user are still applicable for the setting of degraded message sets, essentially creating a whole “class” of algorithms for BEC channels. Three simple examples were chosen to illustrate the main ideas of this class, and it became apparent that the exact algorithmic procedure is mainly determined by the number of sessions and the relation between the message sets rather than the number of users.

APPENDIX A
PROOF OF LEMMA 1

Proof is by induction on time. At the beginning of slot $t = 0$ and since the users have received no information yet, the performed initialization ensures that, for each $\mathcal{S} \subseteq \mathcal{N}$ and $i \in \mathcal{S}$, each packet p stored in $Q_{\mathcal{S}}$ belongs to $\mathcal{D}_i$. Assume now that the inductive hypothesis hold at the beginning of slot t and consider the queue contents at the beginning of slot $t+1$. For each $\mathcal{S} \subseteq \mathcal{N}$ and $i \in \mathcal{S}$, there are only two ways in which a packet p can be stored in queue $Q_{\mathcal{S}}$ at the beginning of slot $t+1$:

- p was already stored in $Q_{\mathcal{S}}$ at the beginning of slot t. Due to the inductive hypothesis, p already has the required property at the beginning of slot t, and therefore slot $t+1$ as well.
- p was not stored in $Q_{\mathcal{S}}$ at the beginning of slot t but was moved to $Q_{\mathcal{S}}$ after the transmission of slot t (so that it appeared in $Q_{\mathcal{S}}$ at the beginning of slot $t+1$). Denoting with $Q_{\mathcal{I}(t)}$ the queue being processed at slot t, an examination of the proposed algorithm reveals that a packet movement is only possible due to step 3 of ACTFB. However, this implies that p can only be moved to these queues $Q_{\mathcal{S}}$ for which $\mathcal{I}(t) \subset \mathcal{S}$ and p was successfully received at slot t by *all* users in $\mathcal{S} - \mathcal{I}(t)$. Clearly then, p satisfies, at the beginning of $t+1$, the required property for all $i \in \mathcal{S} - \mathcal{I}(t)$ (since these users have just received p). Hence, it remains to examine whether p satifies the property for all $i \in \mathcal{I}(t)$.[3] Since p was transmitted during the processing of $Q_{\mathcal{I}(t)}$, it holds $p = \sum_{v \in Q_{\mathcal{T} \supseteq \mathcal{I}(t)}(t)} a_p(v)v$, where we explicitly added the time dependence to show that each packet $v \in Q_{\mathcal{T}}(t)$, with $\mathcal{T} \supseteq \mathcal{I}(t)$, satisfies, due to the inductive hypothesis, the required property for all users in $\mathcal{T}$ (and, therefore, $\mathcal{I}(t)$). We can now use Corollary 1 to conclude that p also satisfies the required property for all $i \in \mathcal{I}(t)$, as a linear combination of packets having this property, to finish the proof.

APPENDIX B
PROOF OF LEMMA 2

The Lemma will be proved after being cast into an equivalent form and a few intermediate results have been established first. Initially, we note that, due to Lemma 1, the packets stored in $Q_{\mathcal{I}}$ at the beginning of slot t, for any $\mathcal{I} \subseteq \mathcal{N}$ and t, can be written in the form of (4). Hence, the set $\{\boldsymbol{b}_p^{(i)} : p \in Q_{\mathcal{I}}(t)\}$ is well defined. Furthermore, consider a packet p transmitted, prior to slot t, while processing $Q_{\mathcal{S}}$ and successfully received by i. We distinguish the following cases.

- $i \in \mathcal{S}$: in this case, Corollary 1 implies that packet $p = \sum_{v \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_p(v)v$ can be written in the form of (4) w.r.t. user i, with $\boldsymbol{b}_p^{(i)} = \sum_{v \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_s(v)\boldsymbol{b}_v^{(i)}$ and $c_p^{(i)} = \sum_{v \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_s(v)c_v^{(i)}$.
- $i \notin \mathcal{S}$: since p was received by i, it can be trivially written in the form of (4), by setting $c_p^{(i)} = p$, $\boldsymbol{b}_p^{(i)} = \mathbf{0}$.

Hence, the set $\{\boldsymbol{b}_p^{(i)} : p \text{ received by user } i \text{ prior to } t\}$ is also well defined, provided that $\boldsymbol{b}_p^{(i)}$ is constructed according to the above case distinction. We now state the following result, which is a stronger version of Lemma 2.

Lemma 3: Under the application of the algorithm, the following condition is true at the *beginning* of each slot t: there exist vector sets $\mathcal{B}_{\mathcal{I}}^{(i)}(t) \subseteq \{\boldsymbol{b}_p^{(i)} : p \in Q_{\mathcal{I}}(t)\}$, for all $\mathcal{I} \subseteq \mathcal{N}$ and $i \in \mathcal{I}$, and $\mathcal{B}_i(t) \subseteq \{\boldsymbol{b}_p^{(i)} : p \text{ received by } i \text{ prior to } t\}$, for all $i \in \mathcal{N}$, such that

- $|\mathcal{B}_{\mathcal{I}}^{(i)}(t)| = T_{\mathcal{I}}^i(t)$, for all $\mathcal{I} \subseteq \mathcal{N}$ and $i \in \mathcal{I}$.
- $\mathcal{B}_i(t) \cup \bigcup_{\substack{\mathcal{I}:\mathcal{I} \subseteq \mathcal{N} \\ T_{\mathcal{I}}^i(t) > 0}} \mathcal{B}_{\mathcal{I}}^{(i)}(t)$ is a basis of $\mathbb{F}_q^{|\mathcal{D}_i|}$ for all $i \in \mathcal{N}$.

It is easy to see why proving Lemma 3 also proves Lemma 2. Specifically, consider the queue and index state at the end of the algorithm (denote the last slot as t_{end}). By the algorithm's construction, it holds $T_{\mathcal{I}}^i(t_{end}+1) = 0$ for all $\mathcal{I} \subseteq \mathcal{N}$ and $i \in \mathcal{I}$, which implies, through the second item in the above list, that $\mathcal{B}_i(t_{end}+1)$ is a basis set of $\mathbb{F}_q^{|\mathcal{D}_i|}$ for all $i \in \mathcal{N}$. Since each element of $\mathcal{B}_i(t_{end}+1)$ corresponds, through (4), to an equation constructed by i upon reception of the corresponding packet p, we conclude that each user has received $|\mathcal{D}_i|$ innovative tokens by the end of the algorithm and Lemma 2 is proved. Hence, in the following we concentrate on proving Lemma 3.

Before proving Lemma 3, we will need some intermediate results. The next proposition is a trivial statement of the union bound for events A_j^c, where ^{c} denotes set complement.

[3]the queues $Q_{\mathcal{S}}$, with $\mathcal{S} \not\supset \mathcal{I}(t)$, remain unchanged between slots t and $t+1$ so that any packets stored there already satisfy the property at the beginning of slot t, and therefore $t+1$.

Proposition 1: For any events A_j, with $j = 1, \dots, m$, it holds

$$\Pr(\cap_{j=1}^m A_j) \geq \sum_{j=1}^m \Pr(A_j) - m + 1.$$

The following result is directly taken from [1].

Lemma 4: Let $\boldsymbol{v}_j$, with $j = 1, \dots, k$, be vectors in vector space $\mathbb{F}_q^M$. Denote $\mathcal{V} = span(\{\boldsymbol{v}_j\}, j = 1, \dots, k)$ and $l = \dim(\mathcal{V})$, with $l \geq 1$. Let α_j, with $j = 1, \dots, k$, be independent random variables uniformly distributed in $\mathbb{F}_q$ and construct the random vector $\boldsymbol{v} = \sum_{j=1}^k \alpha_j \boldsymbol{v}_j$. Then, $\boldsymbol{v}$ is uniformly distributed in $\mathcal{V}$, i.e.

$$\Pr(\boldsymbol{v} = \boldsymbol{e}) = \frac{1}{q^l} \quad \forall\, \boldsymbol{e} \in \mathcal{V}.$$

Additionally, let $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_M\}$ be a basis of $\mathbb{F}_q^M$ and assume that $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_K\} \subseteq \mathcal{V}$ for $1 \leq K \leq M$. It then holds

$$\Pr\left(\{\boldsymbol{v}, \boldsymbol{b}_2, \dots, \boldsymbol{b}_M\} \text{ is basis of } \mathbb{F}_q^M\right) \geq 1 - \frac{1}{q}\ .$$

We are now in position to prove Lemma 3. Consider the degraded message sets $K_1, \dots, K_N$ and denote the standard orthonormal basis of vector space $\mathbb{F}_q^{|\mathcal{D}_i|}$ with $\mathcal{E}_i = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_{|\mathcal{D}_i|}\}$, for $i \in \mathcal{N}$. We define the set

$$\mathcal{E}_i^{(j)} = \left\{\boldsymbol{e}_l \in \mathbb{F}_q^{|\mathcal{D}_i|} : \sum_{m=1}^{j-1} K_m < l \leq \sum_{m=1}^{j} K_m \right\}, \tag{28}$$

so that it holds $\mathcal{E}_i = \biguplus_{j=1}^i \mathcal{E}_i^{(j)}$, where $\biguplus$ denotes a union of disjoint sets. Although the above notation may seem cryptic, it is easily interpreted through the schematic in Fig. 5, which groups the components of a vector in $\mathbb{F}_q^{|\mathcal{D}_i|}$ according to the message sets of interest to user i (i.e. $\mathcal{K}_1, \dots, \mathcal{K}_i$). Hence, the set $\mathcal{E}_i^{(j)}$ contains the orthonormal vectors $\boldsymbol{e}_l$ in $\mathbb{F}_q^{|\mathcal{D}_i|}$ which have a component of 1 in a position belonging to the K_j group.

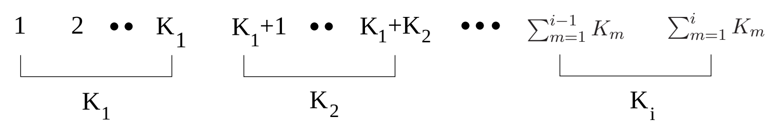

Fig. 5. Component interpretation for any vector in $\mathbb{F}_q^{|\mathcal{D}_i|}$.

The proof of Lemma 3 is, again, by induction on time. Specifically, at the beginning of slot 0, we can construct the following sets, for all $i \in \mathcal{N}$ and $j \in \{i, \dots, N\}$: $\mathcal{B}_{\{i,\dots,N\}}^{(j)}(0) = \mathcal{E}_j^{(i)}$ and $\mathcal{B}_i(0) = \varnothing$ (all other sets $\mathcal{B}_{\mathcal{S}}^{(j)}(0)$ are also set to $\varnothing$). Since it holds, by initialization, $T_{\{i,\dots,N\}}^j = K_i$ and, by construction, $|\mathcal{E}_j^{(i)}| = K_i$, the first condition in Lemma 3 is satisfied. For the second one, we note the following

$$\mathcal{B}_j(0) \cup \bigcup_{\substack{\mathcal{I} \subseteq \mathcal{N} : j \in \mathcal{I} \\ T_{\mathcal{I}}^j(0) > 0}} \mathcal{B}_{\mathcal{I}}^{(j)}(0) = \bigcup_{i=1}^{j} \mathcal{B}_{\{i,\dots,N\}}^{(j)}(0) = \bigcup_{i=1}^{j} \mathcal{E}_j^{(i)} = \mathcal{E}_j. \tag{29}$$

Since $\mathcal{E}_j$ is a basis of $\mathbb{F}_q^{|\mathcal{D}_j|}$, the hypothesis is true for $t = 0$.

We now assume that the inductive hypothesis is true at the beginning of slot t, i.e. there exist sets $\mathcal{B}_{\mathcal{I}}^{(i)}(t)$, $\mathcal{B}_i(t)$ that satisfy the two conditions in Lemma 3. Denote with $Q_{\mathcal{S}}$ the queue being processed at slot t, i.e. the transmitted symbol s at slot t has the form $s = \sum_{p \in Q_{\mathcal{T} \subseteq \mathcal{S}}} a_s(p) p$, and construct the set

$$\mathcal{R}_{\mathcal{S}}(t) = \left\{ i \in \mathcal{S} : \exists\, \mathcal{T} \subseteq \mathcal{S} \text{ s.t. } T_{\mathcal{T}}^i(t) > 0 \right\} \tag{30}$$

The set $\mathcal{R}_{\mathcal{S}}(t)$ is certainly non-empty since there exists at least one $i \in \mathcal{S}$ with $T_{\mathcal{S}}^i(t) > 0$ (otherwise, the processing of $Q_{\mathcal{S}}$ would have been completed prior to slot t, due to step 5 of the algorithm, as explained in Section III-A). Some thought reveals that $\mathcal{R}_{\mathcal{S}}(t)$ represents the set of users i whose $T_{\mathcal{I}}^i$ indices can be potentially modified due to steps 2, 3 of `ACTFB`.

The next result essentially shows that, when processing $Q_{\mathcal{S}}$, the coefficients of the transmitted linear combination s can be selected in such a way that s is an innovative token for all $i \in \mathcal{R}_{\mathcal{S}}(t)$.

Lemma 5: Assume that queue $Q_{\mathcal{S}}$ is processed at slot t and there exist sets $\mathcal{B}_i(t)$, $\mathcal{B}_{\mathcal{I}}^{(i)}(t)$ that satisfy the conditions of Lemma 3 at the beginning of slot t. Define $\mathcal{R}_{\mathcal{S}}(t)$ according to (30) and, for each $i \in \mathcal{R}_{\mathcal{S}}(t)$, pick a set $\hat{\mathcal{S}}_i \supseteq \mathcal{S}$ of smallest cardinality (with an arbitrary tie-breaker) such that $T^i_{\hat{\mathcal{S}}_i}(t) > 0$ and pick an arbitrary $\hat{\boldsymbol{b}}_i \in \mathcal{B}^{(i)}_{\hat{\mathcal{S}}_i}(t)$. Then, for the packet $s = \sum_{p \in Q_{\mathcal{T} \supseteq \mathcal{S}}} a_s(p)p$ transmitted at slot t, we can select the coefficients $a_s(p)$ such that the set $\{\boldsymbol{b}_s^{(i)}\} \cup \mathcal{B}_i(t) \cup \bigcup_{\substack{\mathcal{I}: i \in \mathcal{I} \\ T^i_{\mathcal{I}}(t) > 0}} \mathcal{B}^{(i)}_{\mathcal{I}}(t) - \{\hat{\boldsymbol{b}}_i\}$ is a basis set for $\mathbb{F}^{|\mathcal{D}_i|}$ for all $i \in \mathcal{R}_{\mathcal{S}}(t)$.

Proof of Lemma 5: To make the notation more compact, we denote $\mathring{B}(t) = \mathcal{B}_i(t) \cup \bigcup_{\substack{\mathcal{I}: \mathcal{I} \subseteq \mathcal{N} \\ T^i_{\mathcal{I}}(t) > 0}} \mathcal{B}^{(i)}_{\mathcal{I}}(t)$. Proof is by a standard probabilistic argument. Specifically, we assume that the coefficients are iid randomly and uniformly generated in $\mathbb{F}_q$ and, for each $i \in \mathcal{R}_{\mathcal{S}}(t)$, we define the event $A_i = \left\{\{\boldsymbol{b}_s^{(i)}\} \cup \mathring{B}(t) - \{\hat{\boldsymbol{b}}_i\} \text{ is basis set of } \mathbb{F}_q^{|\mathcal{D}_i|}\right\}$. Since Lemma 4 implies that $\Pr(A_i) \geq 1 - 1/q$, we can apply Proposition 1 to write

$$\Pr\left(\cap_{i \in \mathcal{R}_{\mathcal{S}}(t)} A_i\right) \geq |\mathcal{R}_{\mathcal{S}}(t)| \left(1 - \frac{1}{q}\right) - |\mathcal{R}_{\mathcal{S}}(t)| + 1 \geq 1 - \frac{|\mathcal{R}_{\mathcal{S}}(t)|}{q} \geq 1 - \frac{N}{q}. \tag{31}$$

Clearly, selecting $q > N$ results in a positive probability for the event $\cap_{i \in \mathcal{R}_{\mathcal{S}}(t)} A_i$, which implies that there indeed exist coefficient $a_s(p)$ such that $\{\boldsymbol{b}_s^{(i)}\} \cup \mathring{B}(t) - \{\hat{\boldsymbol{b}}_i\}$ is a basis of $\mathbb{F}_q^{|\mathcal{D}_i|}$ for all $i \in \mathcal{R}_{\mathcal{S}}(t)$. ■

We now revert to the proof of Lemma 3 and assume that the coefficients $a_s(p)$ for the transmitted linear combination s at slot t are selected according to Lemma 5. We will examine the effect of the ACTFB actions at the end of slot t on sets $\mathcal{B}_i(t)$, for $i \in \mathcal{N}$, and $\mathcal{B}^{(i)}_{\mathcal{I}}(t)$, for $\mathcal{I} \subseteq \mathcal{N}$ and $i \in \mathcal{I}$.

Clearly, for each $i \notin \mathcal{R}_{\mathcal{S}}(t)$, it holds either $i \notin \mathcal{S}$ or $i \in \mathcal{S}$ and $T^i_{\mathcal{T}}(t) = 0$ for all sets $\mathcal{T} \supseteq \mathcal{S}$. In this case, ACTFB does not modify any of the $T^i_{\mathcal{I}}$ indices, for any $\mathcal{I} \subseteq \mathcal{N}$. Hence, there is no need to modify the $\mathcal{B}_i(t)$, $\mathcal{B}^{(i)}_{\mathcal{I}}(t)$ sets (i.e. $\mathcal{B}_i(t+1) = \mathcal{B}_i(t)$, and similarly for $\mathcal{B}^{(i)}_{\mathcal{I}}(t)$) and since the inductive hypothesis holds, by assumption, for $i \notin \mathcal{R}_{\mathcal{S}}(t)$ at the beginning of slot t, it trivially holds at the beginning of $t+1$.

We now concentrate on $i \in \mathcal{R}_{\mathcal{S}}(t)$ and consider the following mutually exclusive cases

1) if i receives the transmitted packet s then, according to step 2 of ACTFB, index $T^i_{\hat{\mathcal{S}}_i}$ is reduced by 1, i.e. $T^i_{\hat{\mathcal{S}}_i}(t+1) = T^i_{\hat{\mathcal{S}}_i}(t) - 1$. We accordingly select $\mathcal{B}^{(i)}_{\hat{\mathcal{S}}_i}(t+1) = \mathcal{B}^{(i)}_{\hat{\mathcal{S}}_i}(t) - \{\hat{\boldsymbol{b}}_i\}$ and $\mathcal{B}_i(t+1) = \mathcal{B}_i(t) \cup \{\boldsymbol{b}_s^{(i)}\}$, while all other $\mathcal{B}^{(i)}_{\mathcal{I}}$ sets remain unchanged. Lemma 5 now immediately implies that set $\mathcal{B}_i(t+1) \cup \bigcup_{\substack{\mathcal{I}: \mathcal{I} \subseteq \mathcal{N} \\ T^i_{\mathcal{I}}(t+1) > 0}} \mathcal{B}^{(i)}_{\mathcal{I}}(t)$ is a basis set of $\mathbb{F}_q^{|\mathcal{D}_i|}$, so that the inductive hypothesis is true for $t+1$.
2) if i erases s and the set $\mathcal{G}$ of users that receive s contains at least one user outside $\mathcal{S}$ (i.e. $\mathcal{G} \cap (\mathcal{N} - \mathcal{S}) \neq \varnothing$), then, if $T^i_{\mathcal{S}}(t) > 0$, indices $T^i_{\mathcal{S}}$, $T^i_{\mathcal{S} \cup \mathcal{G}}$ are decreased/increased by 1, respectively. We now select $\mathcal{B}^{(i)}_{\mathcal{S}}(t+1) = \mathcal{B}^{(i)}_{\mathcal{S}}(t) - \{\hat{\boldsymbol{b}}_i\}$ and $\mathcal{B}^{(i)}_{\mathcal{S} \cup \mathcal{G}}(t+1) = \mathcal{B}^{(i)}_{\mathcal{S} \cup \mathcal{G}}(t) \cup \{\boldsymbol{b}_s^{(i)}\}$ while all other sets $\mathcal{B}^{(i)}_{\mathcal{I}}(t)$ remain unchanged. The chosen update of the sets ensures, through Lemma 5, that the conditions in Lemma 3 are satisfied by all $\mathcal{B}_i$, $\mathcal{B}^{(i)}_{\mathcal{I}}$.
3) if i erases s and $\mathcal{G} \subseteq \mathcal{S}$, no T indices are changed and, as a result, the sets $\mathcal{B}_i$, $\mathcal{B}^{(i)}_{\mathcal{I}}$ do not need to be changed. The sets $\mathcal{B}_i$, $\mathcal{B}^{(i)}_{\mathcal{I}}$ already satisfy the conditions in Lemma 3 at the beginning of slot t; hence, they also satisfy them at the beginning of $t+1$.

Since we have examined all possible cases, we conclude that, by selecting the coefficients according to Lemma 5, we can always construct sets $\mathcal{B}_i$, $\mathcal{B}^{(i)}_{\mathcal{I}}$ that satisfy the conditions of Lemma 3 at the beginning of slot $t+1$, so that the induction is complete.

APPENDIX C
ACHIEVABLE REGION OF THE ALGORITHM FOR CASE (B) OF FIG. 1

The total number of slots required by the algorithm is given through (23), (24) as

$$T^* = \sum_{\{N\} \subseteq \mathcal{S} \subset \mathcal{N}} T^*_{\mathcal{S}} + \max\left[\frac{K_1 + k^N_{\mathcal{N}}}{1 - \epsilon_{\{N\}}},\ \max_{i \neq N}\left(\frac{K_1}{1 - \epsilon_{\{i\}}} - \sum_{\{i,N\} \subseteq \mathcal{S} \subset \mathcal{N}} T^*_{\mathcal{S}}\right)\right], \tag{32}$$

Introducing the notation

$$f_{\mathcal{S}}^N = \sum_{\mathcal{H}\subseteq\mathcal{S}-\{N\}} \frac{(-1)^{|\mathcal{S}|-|\mathcal{H}|-1}}{1-\epsilon_{\mathcal{N}-\mathcal{H}}} \tag{33}$$

for all $\mathcal{S}$ with $N \in \mathcal{S}$, we can use (19), (22) to write

$$\begin{aligned} T_{\mathcal{S}}^* &= K_N f_{\mathcal{S}}^N, \\ \frac{k_{\mathcal{N}}^N}{1-\epsilon_{\{N\}}} &= K_N f_{\mathcal{N}}^N. \end{aligned} \tag{34}$$

Hence, placing the term $\sum_{\{N\}\subseteq\mathcal{S}\subset\mathcal{N}} T_{\mathcal{S}}^*$ inside the max in (32) yields

$$T^* = \max\left[\frac{K_1}{1-\epsilon_{\{N\}}} + K_N \sum_{\{N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} f_{\mathcal{S}}^N,\ \max_{i\neq N}\left(\frac{K_1}{1-\epsilon_{\{i\}}} + K_N \sum_{\{N\}\subseteq\mathcal{S}\subset\mathcal{N}} f_{\mathcal{S}}^N - K_N \sum_{\{i,N\}\subseteq\mathcal{S}\subset\mathcal{N}} f_{\mathcal{S}}^N\right)\right] \tag{35}$$

We now use the following result [1, eq. (40)]

$$\sum_{\{N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} f_{\mathcal{S}}^N = \frac{1}{1-\epsilon_{\{N\}}}, \tag{36}$$

which converts the first term in the max of (35) into $\frac{K_1+K_N}{1-\epsilon_{\{N\}}}$ and also yields through (32)–(34)

$$K_N \sum_{\{N\}\subseteq\mathcal{S}\subset\mathcal{N}} f_{\mathcal{S}}^N = \frac{K_N}{1-\epsilon_{\{N\}}} - f_{\mathcal{N}}^N K_N. \tag{37}$$

Inserting the above relation into the second term of the max in (35) yields

$$T^* = \max\left[\frac{K_1+K_N}{1-\epsilon_{\{N\}}},\ \max_{i\neq N}\left(\frac{K_1}{1-\epsilon_{\{i\}}} + \frac{K_N}{1-\epsilon_{\{N\}}} - K_N \sum_{\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} f_{\mathcal{S}}^N\right)\right]. \tag{38}$$

The last sum in the second term is computed through (33) as follows

$$\begin{aligned} \sum_{\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} f_{\mathcal{S}}^N &= \sum_{\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} \sum_{\mathcal{H}\subseteq\mathcal{S}-\{N\}} \frac{(-1)^{|\mathcal{S}|-|\mathcal{H}|-1}}{1-\epsilon_{\mathcal{N}-\mathcal{H}}} = \sum_{\mathcal{H}\subseteq\mathcal{N}-\{N\}} \sum_{\mathcal{H}\cup\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} \frac{(-1)^{|\mathcal{S}|-|\mathcal{H}|-1}}{1-\epsilon_{\mathcal{N}-\mathcal{H}}} \\ &= \sum_{\mathcal{H}\subseteq\mathcal{N}-\{N\}} \frac{(-1)^{|\mathcal{H}|+1}}{1-\epsilon_{\mathcal{N}-\mathcal{H}}} \sum_{\mathcal{H}\cup\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} (-1)^{|\mathcal{S}|}, \end{aligned} \tag{39}$$

where the last equality in the first line is due to a change in the order of summation.

Using the binomial expansion theorem, it is easy to show the following result for arbitrary sets $\mathcal{F}_1$, $\mathcal{F}_2$ with $\mathcal{F}_1 \subseteq \mathcal{F}_2$.

$$\sum_{\mathcal{F}_1\subseteq\mathcal{S}\subseteq\mathcal{F}_2} (-1)^{|\mathcal{S}|} = \begin{cases} 0 & \text{if } \mathcal{F}_1 \subset \mathcal{F}_2, \\ (-1)^{|\mathcal{F}_1|} & \text{if } \mathcal{F}_1 = \mathcal{F}_2. \end{cases} \tag{40}$$

Hence, the only terms that produce a non-zero inner sum in (39) are $\mathcal{H} = \mathcal{N} - \{i, N\}$ and $\mathcal{H} = \mathcal{N} - \{N\}$, which converts (39) into

$$\sum_{\{i,N\}\subseteq\mathcal{S}\subseteq\mathcal{N}} f_{\mathcal{S}}^N = \frac{1}{1-\epsilon_{\{N\}}} - \frac{1}{1-\epsilon_{\{i,N\}}}, \tag{41}$$

Inserting the last expression into (35) yields

$$T^* = \max\left[\frac{K_1+K_N}{1-\epsilon_{\{N\}}},\ \max_{i\neq N}\left(\frac{K_1}{1-\epsilon_{\{i\}}} + \frac{K_N}{1-\epsilon_{\{i,N\}}}\right)\right], \tag{42}$$

which immediately produces the achievable region in (2).

REFERENCES

[1] M. Gatzianas, L. Georgiadis, and L. Tassiulas, “Muliple user broadcast erasure channel with feedback — capacity and algorithms,” submitted to IEEE Trans. Inform. Theory. [Online]. Available: http://arxiv.org/abs/1009.1254

[2] C.-C. Wang, “Capacity of 1–to-K broadcast packet erasure channels with channel output feedback,” in *Proc. 48th Annual Allerton Conference*, October 2010, to appear in IEEE Trans. Inform. Theory. [Online]. Available: http://arxiv.org/abs/1010.2436v1

[3] L. Georgiadis and L. Tassiulas, “Broadcast erasure channel with feedback — capacity and algorithms,” in *Proc. 5th Workshop on Network Coding Theory and Applications*, June 2009, pp. 54–61.

[4] C. Nair and A. E. Gamal, “The capacity region of a class of 3-receiver broadcast channels with degraded message sets,” in *Proc. ISIT*, June 2008.

[5] S. Gheorghiu, S. Saeedi, C. Fragouli, and A. Lopez, “Degraded multicasting with network coding over the combination network,” in *Proc. NetCod 2011*, July 2011.

[6] T. Cover and J. Thomas, *Elements of information theory*, 2nd ed. John Wiley, 2006.

[7] L. Ozarow and S. Leung-Yan-Cheong, “An achievable region and outer bound for the Gaussian broadcast channel with feedback,” *IEEE Trans. Inform. Theory*, vol. 30, no. 4, pp. 667–671, July 1984.

[8] P. Bergmans, “Random coding theorem for broadcast channels with degraded components,” *IEEE Trans. Inform. Theory*, vol. 19, no. 2, pp. 197–207, March 1973.

[9] A. E. Gamal, “The feedback capacity of degraded broadcast channels,” *IEEE Trans. Inform. Theory*, vol. 24, no. 3, pp. 379–381, May 1978.

[10] A. Dana and B. Hassibi, “The capacity region of multiple input erasure broadcast channels,” in *Proc. International Symposium on Information Theory (ISIT)*, September 2005, pp. 2315–2319.